AMENDMENT TO THE SPECIFICATION

Please replace the paragraph starting on line 20 of page 2 with the following paragraph.

Matched filters and RACH detectors typically require very high million instructions per second (MIPS) computation power. Depending on the parameters, the matched filter or the RACH detector can require execution of tens of billions or hundreds of billions of operations per second. The high MIPS computation power requirement of matched filters and RACH detectors exceed the capability of a single standard digital signal processing (DSP) processor. The use of a specialized DSP processor or multiple standard DSP processors to perform the computations required costs that were high and undesirable. Conventional usage of Programmable Logic Devices (PLDs) to perform the computations required the PLDs to include a large number of logic elements for the computations and a large number of registers for storing data and results from the matched filter or RACH. This translated into the need for additional real estate and costs that were also undesirable.

Please replace the paragraph starting on line 4 of page 3 with the following paragraph.

Thus, what is needed is a method and apparatus for implementing matched filters and RACH detectors to process code sequences and transmitted samples that is efficient and cost effective.

Please replace the paragraph starting on line 17 of page 10 with the following paragraph. sample sequence group.

The PPT correlator 100 includes a plurality of sample sequence registers 140. The sample sequence registers 140 are coupled to the receiver 130. The sample sequence registers

140 receive a plurality of sample values from the receiver 130. The sample sequence registers 140 may store sample values from a plurality of d sample sequences that are processed in parallel where all d sample sequences include sample values that are in each of the d sample sequences. The sample sequence registers 140 may store sample values from one complete sample sequence group from each of the plurality of d sample sequences at a time for processing during each clock cycle. The sample sequence registers 140 may be implemented by a register or other storage device.

Please replace the paragraph starting on line 25 of page 12 with the following paragraph.

The samples received and stored in receiver 130 (shown in Figure 1) may be organized into a x+1-L contiguous sample sequences, where L is the number of coefficients in a coefficient sequence and x is the number of sample values in the sample. Each of the sample sequences includes a contiguous subset of L sample values from the samples. According to an embodiment of the present invention, a first value in a first sample sequence includes a first value in the sample and each consecutive sample sequence includes a next contiguous value in the sample as a first value in the consecutive sample sequence. Each sample sequence may also be organized into a plurality of contiguous sample sequence groups having n values each. The sample sequence registers 140 (shown in Figure 1) store sample values from d sample sequences groups that are to be processed in parallel during a current clock cycle of the PPT correlator 100. Blocks 211-219 represent the sample sequence registers 140. In this example, the sample sequence registers 211-219 are configured to store n+d-1 sample values where d is 4. Registers 211-216 may store sample values from a first sample sequence group. Registers 212-217 may store sample values from a second sample sequence group. Registers 213-218 may store sample values from a third sample sequence group. Registers 214-219 may store sample values from a fourth sample sequence group.

Please replace the paragraph starting on line 13 of page 19 with the following paragraph.

Figure 6 also illustrates a plurality of samples having the sample values 5 2 –1 3 6 –6 3 –3 5 9 –8 –8 7 [–]6. The sample values may be received and stored in receiver 130 (shown in Figure 1). In this example, the sample has x contiguous sample values where x equals 14. The sample values may be organized into x+1-L contiguous sample sequences where each of the sample sequences includes a contiguous subset of L sample values from the sample. According to an embodiment of the present invention, a first sample value in a first sample sequence includes a first sample value in the sample and each consecutive sample sequence includes a next contiguous sample value in the sample as a first value in the consecutive sample sequence. In this example, a first sample sequence includes the sample values 5 2 –1 3 6 –6. A second sample sequence includes the sample values 2 –1 3 6 –6 3. A third sample sequence includes the sample values –1 3 6 –6 3 –3. A fourth sample sequence includes the sample values 3 6 –6 3 –3 5. A fifth sample sequence includes the sample values 6 –6 3 –3 5 9. A sixth sample sequence includes the sample values –6 3 –3 5 9 –8. A seventh sample sequence includes the sample values 3 –3 5 9 –8 –8. An eighth sample sequence includes the sample values –3 5 9 –8 –8 7. A ninth sample sequence includes the sample values 5 9 –8 –8 7 6.